



Linux based 3G Specification

Multimedia Mobile Phone API

Short Message Service

Document: CELF_MPP_SMS_D_2.2_20051121

WARNING : This is a working draft for review only, it is NOT a published specification of the CE Linux Forum. It is likely that further substantial changes will be made in the course of review and issue resolution. Send comments on this version to:

MppApiComments@tree.celinuxforum.org

Revision History

Revision	Comment	Reviewer	Editor	Date
2.2	Conversion to new format and standards	NEC	AK	05/11/21

DRAFT

0.	<i>Introduction</i>	4
	References	4
0.1.1	Normative	4
0.1.2	Informative	4
1.	<i>Primitives</i>	5
1.1	Data Types and Structures	5
1.1.1	Name	Error! Bookmark not defined.
1.2	Constants	8
1.2.1	Name	8
2.	<i>Functions</i>	Error! Bookmark not defined.
2.1	Symbol: <name>	13
2.1.1	Syntax	13
2.1.2	Argument	13
2.1.3	Return Value	14
2.1.4	Include File	14
2.1.5	Functional Description	14

0. Introduction

<Introduction to this service API>

References

0.1.1 Normative

0.1.2 Informative

DRAFT

1. Primitives

This section contains the definitions of the data types and constants used in the interfaces of this service.

1.1 Data Types and Structures

1.1.1 Event Structure

```
typedef struct {
    int          category ;    //The value is SMSNotify.
    int          subtype;      //The notification event. See below.
    int          info ;        //Unused
    int          subinfo ;     //Unused
    union {
        ...
        <SMS message structure> ; //See the following structure members
                                     //corresponding to subtype.
        ...
    } data ;
} CELF_MP_SM_EVENT;
```

1.1.2 CELF_MP_SM_CLASS

The event classes are defined as follows:

CELF_MP_SM_CLASS_MO_DATA:	SMS-MO send indication notification
CELF_MP_SM_CLASS_MT_DATA:	SMS-MT receive indication notification
CELF_MP_SM_CLASS_STORE:	SMS message-stored notification
CELF_MP_SM_CLASS_ERROR:	SMS error notification
CELF_MP_SM_CLASS_REPORT:	SMS delivery report received indication notification
CELF_MP_SM_CLASS_CNMA:	SMS receive response accepted notification
CELF_MP_SM_CLASS_MEM_AVAILABLE:	SMS receive memory check response notification
CELF_MP_SM_CLASS_ALL:	All notifiedSet of events to be reported

1.1.3 CELF_MP_SM_SND_DATA

Description: SMS send data structure

Definition:

```
typedef struct {
    unsigned char    callref;    //Call reference. Set to zero.
    CELF_MP_SM_RP_ADDRESS
```

```

                                sca;          //Destination service center address
    CELF_MP_SM_SUBMIT  tpdu;        //TPDU data (SMS-SUBMIT)
} CELF_MP_SM_SND_DATA;

```

1.1.4 CELF_MP_SM_RP_ADDRESS

Description: SMS address structure

Definition:

```

typedef struct {
    unsigned char    len;          //Address length    from 0 to 11
    unsigned char    type;        //Address type
    unsigned char    address[CELF_MP_SM_RP_ADDRESS_MAX];
                                //Address data    10 bytes
} CELF_MP_SM_RP_ADDRESS;

```

- The address length is (a) the size of the address type, that is one byte, plus
(b) the size of the valid address data in the address data.

- The address data is separated by nibbles.

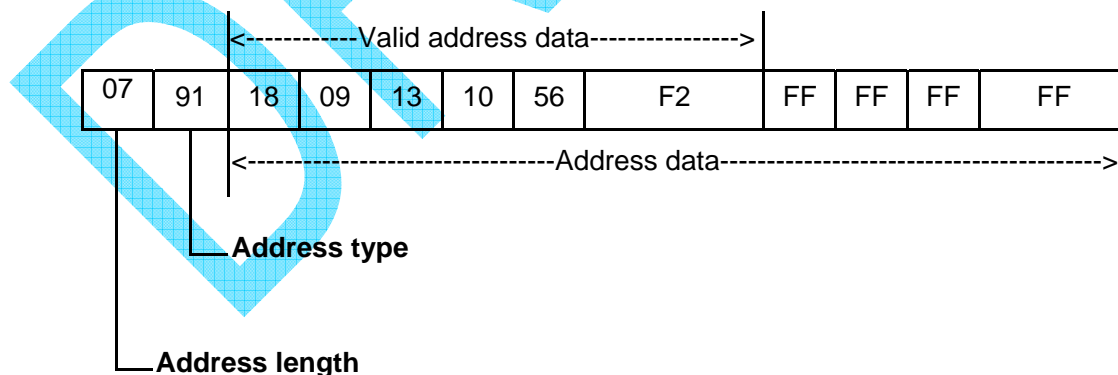
The valid data is filled, and the rest is filled by 0xF.

- Example

Address type: 0x91 (International number, ISDN)

Address data: 81903101652

The memory image of the address above is shown below.



1.1.5 CELF_MP_SM_RCV_RES

Description: SMS receive data structure

Definition:

```
typedef struct {  
    unsigned char    callref;    //Call reference. same value specified by the  
                                //received event  
  
    CELF_MP_SM_DELIVER_ACK  
                                tpd;    //TPDU data (SMS-DELIVER-REPORT for RP-  
                                //ACK)  
  
} CELF_MP_SM_RCV_RES;
```

1.1.6 CELF_MP_SM_RCV_ERR_RES

Description: SMS receive failure structure

Definition:

```
typedef struct {  
    unsigned char    callref;    //Call reference. same value specified by the  
                                //received event  
  
    CELF_MP_SM_DELIVER_ERROR  
                                tpd;    //TPDU data (SMS-DELIVER-REPORT for RP-  
                                //ERROR)  
  
} CELF_MP_SM_RCV_ERR_RES;
```

1.1.7 CELF_MP_SM_RCV_MEM_CHK

Description: SMS receive memory check structure

Definition:

```
typedef struct {  
    unsigned char    callref;    //Call reference. Set to zero.  
} CELF_MP_SM_RCV_MEM_CHK;
```

1.1.8 CELF_MP_SM_ABORT

Description: SMS abort request structure

Definition:

```
typedef struct {  
    unsigned char    callref;    //Call reference. Set to zero.  
} CELF_MP_SM_ABORT;
```

1.2 Constants

1.2.1 Name

Description:

Definition:

DRAFT

1.3 To be clarified

Value of Subtype	Type of SMS message structure	Member	Refer to
SMSNotify_MO_DATA_IND (SMS-MO send indication notification)	CELF_MP_SM_CMGS_RES	cmgs_res	(1)
SMSNotify_MT_DATA_IND (SMS-MT receive indication notification)	CELF_MP_SM_CMTI_RES	cmti_res	(2)
SMSNotify_STORE_IND (SMS message-stored notification)	CELF_MP_SM_CMTI_RES	cmti_res	(2)
SMSNotify_MEMORY_AVAILABLE_IND (SMS receive memory check response notification)	CELF_MP_SM_MEM_AVAILABLE_RES	mem_avail_res	(5)
SMSNotify_ERROR_IND (SMS Error indication notification)	CELF_MP_SM_CMSERR_RES	cmserr_res	(3)
SMSNotify_REPORT_IND (SMS delivery report received indication notification)	CELF_MP_SM_CMTI_RES	cmti_res	(2)
SMSNotify_CNMA_IND (SMS receive response accepted notification)	CELF_MP_SM_CNMA_RES	cnma_res	(4)

1.3.1.1 (3) Message structure

(i) CMGS-Res

```
typedef struct {
    unsigned char    cr;           Call reference
    unsigned char    mr;           Message reference See "3G TS 23.040".
    union {
        CELF_MP_SM_SUBMIT_ACK
        ack;           TPDU data (SMS-SUBMIT-REPORT for RP-ACK)
```

See "3G TS 23.040".

```

    } tpdu;
} CELF_MP_SM_CMGS_RES ;

```

(ii) CMTI-Res

```

typedef struct {
    unsigned char    cr;           Call reference
    unsigned char    e_mr;        Message reference flag (0: Disabled, 1:
Enabled)
    unsigned char    mr;           Message reference
    unsigned char    e_rpsc;      Service center address flag (0: Disabled, 1:
Enabled)
    CELF_MP_SM_RP_ADDRESS
        rpsc;                     Service center address
    union {
        CELF_MP_SM_DELIVER
            deliver;               TPDU data (SMS-DELIVER)
                                   See "3G TS 23.040".
        CELF_MP_SM_REPORT
            status;               TPDU data (SMS-STATUS-REPORT)
                                   See "3G TS 23.040".
    } tpdu;
} CELF_MP_SM_CMTI_RES;

```

(iii) MEM_AVAIL_Res

```

typedef struct tagCELF_MP_SM_MEM_AVAIL_RES {
    unsigned char    cr;           Call reference
    unsigned char    mr;           Message reference
    unsigned char    msg_type;     Message type    It is indicates the type of
received message.
                                   3: avairable *1
                                   5: avairable *2
    unsigned char    e_cme;        Error information flag (0: Disabled, 1: Enabled)
    CELF_MP_SM_CME
        cme;                      Error information
                                   See "3G TS 23.040" and "3G TS 24.011".
    union {

```

CELF_MP_SM_SUBMIT_ACK

ack; TPDU data (SMS-SUBMIT-REPORT for RP-ACK)
See "3G TS 23.040".

CELF_MP_SM_SUBMIT_ERROR

error ; TPDU data (SMS-SUBMIT-REPORT for RP-
ERROR)
See" 3G TS 23.040".

} tpdu;
} CELF_MP_SM_MEM_AVAIL_RES ;

(iv) CMSERR-Res

typedef struct {
unsigned char cr;

Call reference
120 Error at sending
128 to 255 Error at reception

CELF_MP_SM_CME

cme; Error information
See "3G TS 23.040" and "3G TS 24.011".

} CELF_MP_SM_CMSERR_RES;

(v) CNMA-Res

typedef struct {
unsigned char cr;

Call reference
Set the call ID number having been set for the

SMS-MT receive response

or SMS-MT receive failure response.

unsigned char msg_type;
send from.

Message type It indicates the message type to

2: [RESPOND SMS-MT RECEIVE](#)

4: [RESPOND SMS-MT FAILURE RECEIVE](#)

} CELF_MP_SM_CNMA_RES;

typedef struct tag CELF_MP_SM_MEM_AVAIL_RES {

unsigned char cr; Call reference
unsigned char mr; Message reference
unsigned char msg_type; Message type It is indicates the type of
received message.

3: available *1

5: available *2

unsigned char e_cme; Error information flag (0: Disabled, 1: Enabled)

CELF_MP_SM_CME cme; Error information See "3G TS 23.040"

and "3G TS 24.011".

union {

CELF_MP_SM_SUBMIT_ACK

ack; TPDU data (SMS-SUBMIT-REPORT for RP-ACK)

See "3G TS 23.040".

CELF_MP_SM_SUBMIT_ERROR

error; TPDU data (SMS-SUBMIT-REPORT for RP-

ERROR)

See "3G TS 23.040".

} tpd;

} CELF_MP_SM_MEM_AVAIL_RES ;

2. Start Notification

2.1 Symbol: *celf_mp_sm_notification_start*

2.1.1 Syntax

```
CelfMpCallback celf_mp_sm_notification_start (  
    celfMpAppId app_id,  
    celfMpSmNotifySet event_set,  
    celfMpCallback callback_func );
```

2.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: event_set

Type: celfMpSmNotifySet

I/O: I

Description:

Notification event set. Events that are classified as belonging to one of the `CelfMpSmNotifySet` class **may** be registered to have a callback function called when the event occurs for the application identified by `app_id`. Classes of events are enabled by setting the corresponding bit in `event_set`:

The event classes are defined as follows:

CELF_MP_SM_CLASS_MO_DATA:	SMS-MO send indication notification
CELF_MP_SM_CLASS_MT_DATA:	SMS-MT receive indication notification
CELF_MP_SM_CLASS_STORE:	SMS message-stored notification
CELF_MP_SM_CLASS_ERROR:	SMS error notification
CELF_MP_SM_CLASS_REPORT:	SMS delivery report received indication notification
CELF_MP_SM_CLASS_CNMA:	SMS receive response accepted notification
CELF_MP_SM_CLASS_MEM_AVAILABLE:	SMS receive memory check response notification
CELF_MP_SM_CLASS_ALL:	All notifiedSet of events to be reported

A callback **may** be registered for all classes of events using special event class

`CELF_MP_SM_CLASS_ALL`, however to reduce overhead it is recommended that only the needed event classes **should** be registered.

Name: callback_func

Type: CelfMpCallback

I/O: I

Description:

The callback function, which **shall** be called when an event occurs from one of the classes in `event_set`.

2.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_notification_start()` **shall** return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_EVENT_SET_ERR:	Notification event set is not valid
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

2.1.4 Include File

`/usr/include/celf/mp_sm.h`

2.1.5 Functional Description

This function starts the monitoring the SMS communication status.

The occurrence of the event is notified to the application, specified by `app_id`.

Events to be notified are described in section 0.1.

3. Stop Notification

3.1 Symbol: *celf_mp_sm_notification_stop*

3.1.1 Syntax

```
CelfMpCallback celf_mp_sm_notification_stop (  
    celfMpAppId app_id,  
    celfMpSmNotifySet event_set );
```

3.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: event_set

Type: celfMpSmNotifySet

I/O: I

Description:

Notification event set. Events that are classified as belonging to one of the `CelfMpSmNotifySet` class **may** be registered to have a callback function called when the event occurs for the application identified by `app_id`. Classes of events are enabled by setting the corresponding bit in `event_set`:

The event classes are defined as follows:

<code>CELf_MP_SM_CLASS_MO_DATA:</code>	SMS-MO send indication notification
<code>CELf_MP_SM_CLASS_MT_DATA:</code>	SMS-MT receive indication notification
<code>CELf_MP_SM_CLASS_STORE:</code>	SMS message-stored notification
<code>CELf_MP_SM_CLASS_ERROR:</code>	SMS error notification
<code>CELf_MP_SM_CLASS_REPORT:</code>	SMS delivery report received indication notification
<code>CELf_MP_SM_CLASS_CNMA:</code>	SMS receive response accepted notification
<code>CELf_MP_SM_CLASS_MEM_AVAILABLE:</code>	SMS receive memory check response notification
<code>CELf_MP_SM_CLASS_ALL:</code>	All notifiedSet of events to be reported

A callback **may** be registered for all classes of events using special event class

`CELf_MP_SM_CLASS_ALL`, however to reduce overhead it is recommended that only the needed event classes **should** be registered.

3.1.3 Return Value

Type: CelfMpStatus

I/O: 0

Description:

`celf_mp_sm_notification_start()` **shall** return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_EVENT_SET_ERR:	Notification event set is not valid
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

3.1.4 Include File

`/usr/include/celf/mp_sm.h`

3.1.5 Functional Description

This function stops the notifying of the event about specified SMS communication.

Events to be notified are described in section 0.1.

4. Send SMS

4.1 Symbol: *celf_mp_sm_send*

4.1.1 Syntax

```
CelfMpCallback celf_mp_sm_send (  
    celfMpAppId app_id,  
    CELF_MP_SM_SND_DATA send_data );
```

4.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: send_data

Type: CELF_MP_SM_SND_DATA*

I/O: I

Description:

Address to the SMS send data structure.

4.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_send()` shall return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_PARAM_ERR:	Parameter Error.
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

4.1.4 Include File

`/usr/include/celf/mp_sm.h`

4.1.5 Functional Description

This function sends the SMS data.

If the SMS is sent successfully, the following event is notified.

SMS-MO sends indication notification (SELF_MP_SM_CLASS_MO_DATA)

If the SMS sending is not successful, the following error event is notified.

SMS error notification (SELF_MP_SM_CLASS_ERROR)

This function does not validate nor convert the TPDU data content.

DRAFT

5. Receive SMS

5.1 Symbol: *celf_mp_sm_receive_respond*

5.1.1 Syntax

```
CelfMpCallback celf_mp_sm_receive_respond (  
    celfMpAppId app_id,  
    CELF_MP_SM_RCV_RES rcv_res );
```

5.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: rcv_res

Type: CELF_MP_SM_RCV_RES *

I/O: I

Description:

Address to the SMS receive data structure.

5.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_receive_respond()` shall return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_PARAM_ERR:	Parameter Error.
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

5.1.4 Include File

`/usr/include/celf/mp_sm.h`

5.1.5 Functional Description

This function reports to the MPS that the data is received successfully by each event.

Classification: Short Message Service

The event is SMS-MT receive indication notification (CELF_MP_SM_CLASS_MT_DATA),
SMS message-stored notification (CELF_MP_SM_CLASS_STORE) or
SMS delivery report received indication notification (CELF_MP_SM_CLASS_REPORT).
When this response is completed, the event SMS receive response accepted notification
(CELF_MP_SM_CLASS_CNMA) is notified.

This function does not validate nor convert the TPDU data content.

DRAFT

6. Receive SMS failure

6.1 Symbol: *celf_mp_sm_failure_respond*

6.1.1 Syntax

```
CelfMpCallback celf_mp_sm_failure_respond (  
    celfMpAppId app_id,  
    CELF_MP_SM_RCV_ERR_RES rcv_err_res );
```

6.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: rcv_err_res

Type: CELF_MP_SM_RCV_ERR_RES *

I/O: I

Description:

Address to the SMS receive failure structure.

6.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_failure_respond()` shall return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_PARAM_ERR:	Parameter Error
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

6.1.4 Include File

`/usr/include/celf/mp_sm.h`

6.1.5 Functional Description

This function reports to the MPS that the data is received unsuccessfully by each event.

Classification: Short Message Service

The event is SMS-MT receive indication notification (CELF_MP_SM_CLASS_MT_DATA), SMS message-stored notification (CELF_MP_SM_CLASS_STORE) or SMS delivery report received indication notification (CELF_MP_SM_CLASS_REPORT). When this response is completed, the event SMS receive response accepted notification (CELF_MP_SM_CLASS_CNMA) is notified.

This function does not validate nor convert the TPDU data content.

DRAFT

7. Receive SMS memory check

7.1 Symbol: *celf_mp_sm_mem_check*

7.1.1 Syntax

```
CelfMpCallback celf_mp_sm_mem_check (  
    celfMpAppId app_id,  
    CELF_MP_SM_RCV_MEM_CHK rcv_mem_chk );
```

7.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: rcv_mem_chk

Type: CELF_MP_SM_RCV_MEM_CHK *

I/O: I

Description:

Address to the SMS memory check structure.

7.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_mem_check()` shall return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_PARAM_ERR:	Parameter Error
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

7.1.4 Include File

`/usr/include/celf/mp_sm.h`

7.1.5 Functional Description

This function sends the information to the Message Processing System(MPS) that the SMS-MT reception is enabled.

If the SMS is sent successfully the following event is notified:

SMS receive memory check response notification
(SELF_MP_SM_CLASS_MEMORY_AVAILABLE)

If the SMS sending is not successful, the following error event is notified:

SMS error notification (SELF_MP_SM_CLASS_ERROR)

This function does not validate nor convert the TPDU data content.

DRAFT

8. Abort SMS communication

8.1 Symbol: *celf_mp_sm_abort*

8.1.1 Syntax

```
CelfMpCallback celf_mp_sm_abort (  
    celfMpAppId app_id,  
    CELF_MP_SM_ABORT abort ) ;
```

8.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

Name: abort

Type: CELF_MP_SM_ABORT *

I/O: I

Description:

Address to the SMS abort request structure.

8.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_abort ()` shall return one of the following values:

CELF_MP_STATUS_OK:	successful completion
CELF_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELF_MP_STATUS_PARAM_ERR:	Parameter Error
CELF_MP_STATUS_ERR:	Other unsuccessful completion.

8.1.4 Include File

`/usr/include/celf/mp_sm.h`

8.1.5 Functional Description

This function requests to abort the communication processing.

9. Get SMS communication status

9.1 Symbol: *celf_mp_sm_status*

9.1.1 Syntax

```
CelfMpCallback celf_mp_sm_status (  
    celfMpAppId app_id);
```

9.1.2 Argument

Name: app_id

Type: celfMpAppId

I/O: I

Description:

Application ID returned from `celf_mp_af_get_app_id()` call.

9.1.3 Return Value

Type: CelfMpStatus

I/O: O

Description:

`celf_mp_sm_abort()` shall return one of the following values:

CELf_MP_SM_COM_STATUS_EMPTY:	Communication terminated (idle)
CELf_MP_SM_COM_STATUS_ACTIVE:	Active
CELf_MP_SM_COM_STATUS_SND:	Sending
CELf_MP_SM_COM_STATUS_RCV:	Receiving
CELf_MP_STATUS_OK:	successful completion
CELf_MP_STATUS_APP_ID_ERR:	Application ID is not valid.
CELf_MP_STATUS_PARAM_ERR:	Parameter Error
CELf_MP_STATUS_ERR:	Other unsuccessful completion.

9.1.4 Include File

`/usr/include/celf/mp_sm.h`

9.1.5 Functional Description

This function acquires the current SMS communication status.

This function allows getting the status of SMS communication without calling “`celf_mp_sm_notification_start()`”.

DRAFT